Fortnightly eFolio Submission – Studio 10

Student Name and ID

Hao Xu (32767919)

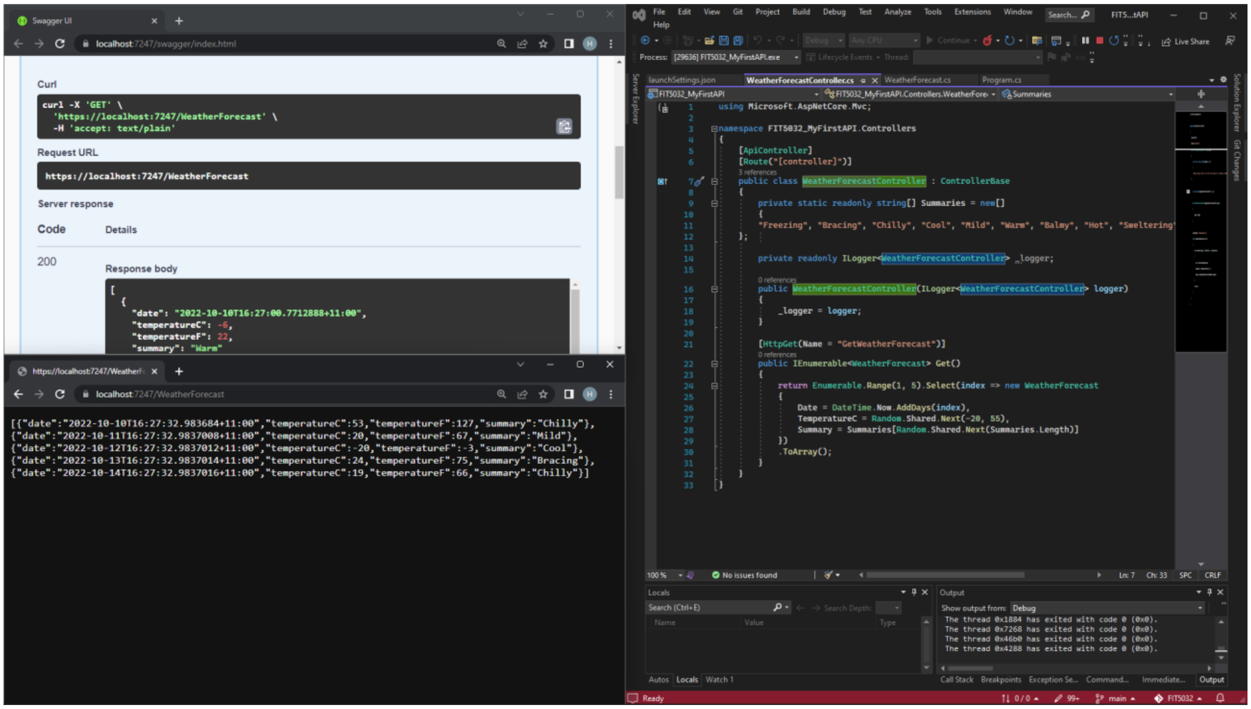
Self-Evaluation

High Distinction

Task 10.1 (First Web API with .NET Core)

GitHub Link: https://github.com/BrantleyXU/FIT5032/tree/main/week10/FIT5032_MyFirstAPI

The following screenshot shows a successfully launched simple application integrated with web API as same as STEP 6 shown in the Week 10 tutorial specification. After executing GET request, the web browser could receive responses containing weather information.



```
curl -X 'GET' \
  'https://localhost:7247/WeatherForecast' \
  -H 'accept: text/plain'
```

```
https://localhost:7247/WeatherForecast
```

```
[
  {
    "date": "2022-10-10T16:27:00.7712888+11:00",
    "temperatureC": -6,
    "temperatureF": 22,
    "summary": "Warm"
```

```
[{"date":"2022-10-10T16:27:32.983684+11:00","temperatureC":53,"temperatureF":127,"summary":"Chilly"},
{"date":"2022-10-11T16:27:32.9837008+11:00","temperatureC":20,"temperatureF":67,"summary":"Mild"},
{"date":"2022-10-12T16:27:32.9837012+11:00","temperatureC":-20,"temperatureF":-3,"summary":"Cool"},
{"date":"2022-10-13T16:27:32.9837014+11:00","temperatureC":24,"temperatureF":75,"summary":"Bracing"},
{"date":"2022-10-14T16:27:32.9837016+11:00","temperatureC":19,"temperatureF":66,"summary":"Chilly"}]
```

```
using Microsoft.AspNetCore.Mvc;

namespace FIT5032_MyFirstAPI.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries = new[]
        {
        "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering"
    };

        private readonly ILogger<WeatherForecastController> _logger;

        public WeatherForecastController(ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }

        [HttpGet(Name = "GetWeatherForecast")]
        public IEnumerable<WeatherForecast> Get()
        {
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
            {
                Date = DateTime.Now.AddDays(index),
                TemperatureC = Random.Shared.Next(-20, 55),
                Summary = Summaries[Random.Shared.Next(Summaries.Length)]
            })
            .ToArray();
        }
    }
}
```

Show output from: Debug

```
The thread 0x1884 has exited with code 0 (0x0).
The thread 0x7268 has exited with code 0 (0x0).
The thread 0x46b0 has exited with code 0 (0x0).
The thread 0x4288 has exited with code 0 (0x0).
```

Figure 1: First Simple Web API

Task 10.2 (Simple AngularJS Application)

GitHub Link:
https://github.com/BrantleyXU/FIT5032/tree/main/week10/FIT5032_SimpleAngularApp

CodePen Link: https://codepen.io/BrantleyXu/pen/wvjYdjO

The following 2 screenshots show a successful simple AngularJS application integrated with .NET Core Web API. The steps and explanations are listed below.

1. At the browser side, I use HTML and JS code which are provided in the tutorial material. And I change the application URL to match my local host application URL.

2. At the server side (Visual Studio), I create end points to match the JS code:

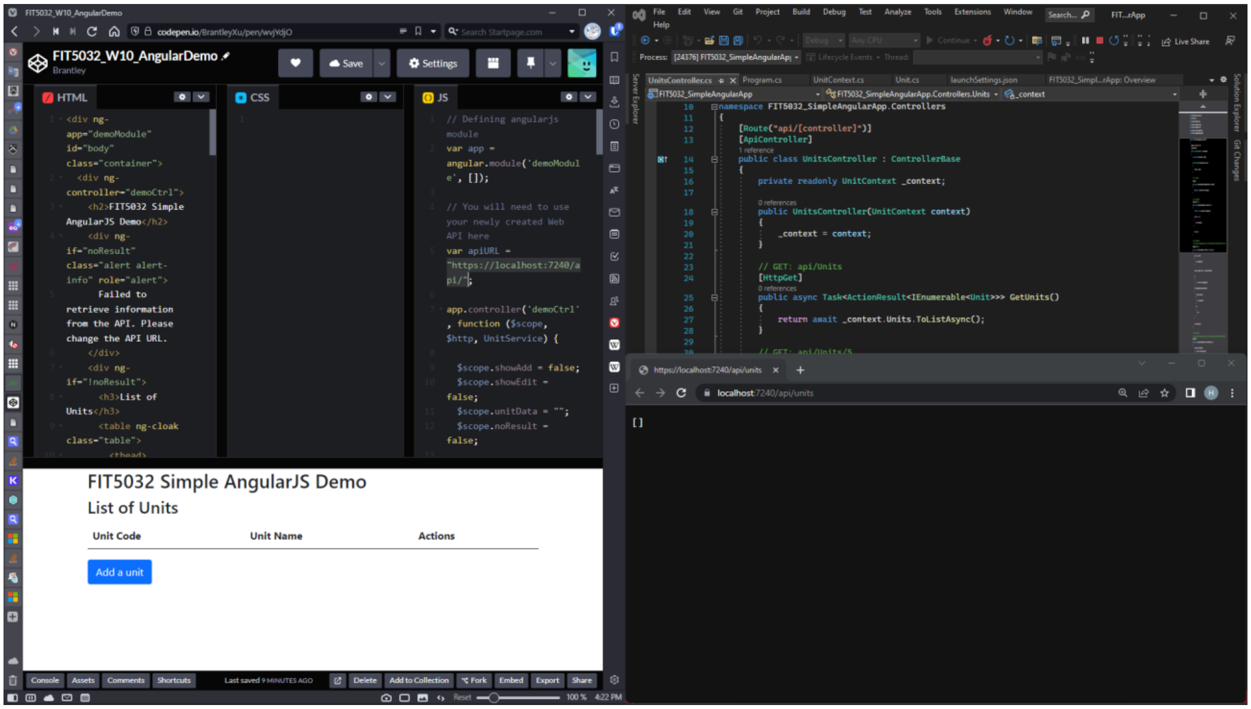
   2.1. I create a model called Unit.cs, with 3 attributes, Id, UnitCode and UnitName;

   2.2. I create a "In-Memory" database called UnitContext.cs to store data. And I register the database in Program.cs;

   2.3. I scaffold the related controller called UnitsController.cs;

   2.4. I enable CORS (Cross-Origin Requests) in my application, in Program.cs file, so that my browser can talk to my server properly.

By doing so, I could successfully execute both JS and ASP.NET code, with no data yet (Figure 2). After I add a unit, the unit can be GET successfully (Figure 3), meaning that the application can work well.



```
<div ng-app="demoModule" id="body" class="container">
  <div ng-controller="demoCtrl">
    <h2>FIT5032 Simple AngularJS Demo</h2>
    <div ng-if="noResult" class="alert alert-info" role="alert">
        Failed to retrieve information from the API. Please change the API URL.
    </div>
    <div ng-if="!noResult">
      <h3>List of Units</h3>
      <table ng-cloak class="table">
        <thead>
```

```
// Defining angularjs module
var app = angular.module('demoModule', []);

// You will need to use your newly created Web API here
var apiURL = "https://localhost:7240/api/";

app.controller('demoCtrl', function ($scope, $http, UnitService) {

  $scope.showAdd = false;
  $scope.showEdit = false;
  $scope.unitData = "";
  $scope.noResult = false;
```

FIT5032 Simple AngularJS Demo

List of Units

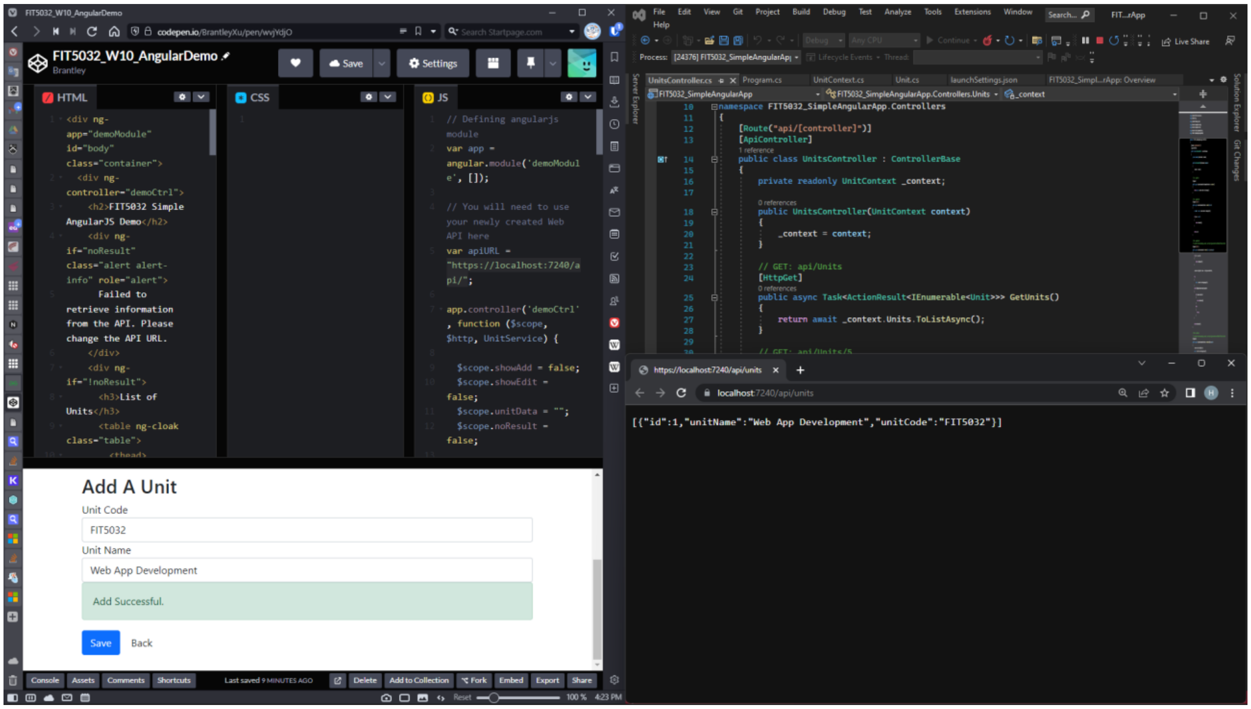
Unit Code	Unit Name	Actions

Add a unit

```
namespace FIT5032_SimpleAngularApp.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class UnitsController : ControllerBase
    {
        private readonly UnitContext _context;

        public UnitsController(UnitContext context)
        {
            _context = context;
        }

        // GET: api/Units
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Unit>>> GetUnits()
        {
            return await _context.Units.ToListAsync();
        }

        // GET: api/Units/5
```

```
[]
```

Figure 2: Initial State



```html
<div ng-app="demoModule" id="body" class="container">
  <div ng-controller="demoCtrl">
    <h2>FIT5032 Simple AngularJS Demo</h2>
    <div ng-if="noResult" class="alert alert-info" role="alert">
      Failed to retrieve information from the API. Please change the API URL.
    </div>
    <div ng-if="!noResult">
      <h3>List of Units</h3>
      <table ng-cloak class="table">
        <thead>
```

```javascript
// Defining angularjs module
var app = angular.module('demoModule', []);

// You will need to use your newly created Web API here
var apiURL = "https://localhost:7240/api/";

app.controller('demoCtrl', function ($scope, $http, UnitService) {

  $scope.showAdd = false;
  $scope.showEdit = false;
  $scope.unitData = "";
  $scope.noResult = false;
```

Add A Unit

Unit Code: FIT5032

Unit Name: Web App Development

Add Successful.

```csharp
namespace FIT5032_SimpleAngularApp.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class UnitsController : ControllerBase
    {
        private readonly UnitContext _context;

        0 references
        public UnitsController(UnitContext context)
        {
            _context = context;
        }

        // GET: api/Units
        [HttpGet]
        0 references
        public async Task<ActionResult<IEnumerable<Unit>>> GetUnits()
        {
            return await _context.Units.ToListAsync();
        }

        // GET: api/Units/5
```

```
[{"id":1,"unitName":"Web App Development","unitCode":"FIT5032"}]
```

Figure 3: Add A Unit State

References

George. (2021, July 12). CORS (3), Enable CORS In .NET Core Web API. *C# Corner*. https://www.c-sharpcorner.com/article/enable-cors-consume-web-api-by-mvc-in-net-core-4/

Microsoft. (2022, June 04). Enable Cross-Origin Requests (CORS) in ASP.NET Core. *Microsoft*. https://learn.microsoft.com/en-us/aspnet/core/security/cors?view=aspnetcore-6.0

Microsoft. (2022, October 06). Tutorial: Create a web API with ASP.NET Core. *Microsoft*. https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio